



# An Accelerator-Based Wireless Sensor Network Processor in 130 nm CMOS

## Citation

Hempstead, Mark, David Brooks, and Gu-Yeon Wei. 2011. "An Accelerator-Based Wireless Sensor Network Processor in 130 Nm CMOS." IEEE J. Emerg. Sel. Topics Circuits Syst. 1 (2) (June): 193–202. doi:10.1109/jetcas.2011.2160751.

## Published Version

doi:10.1109/JETCAS.2011.2160751

## Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:27770104>

## Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP>

## Share Your Story

The Harvard community has made this article openly available.  
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

# An accelerator-based wireless sensor network processor in 130nm CMOS

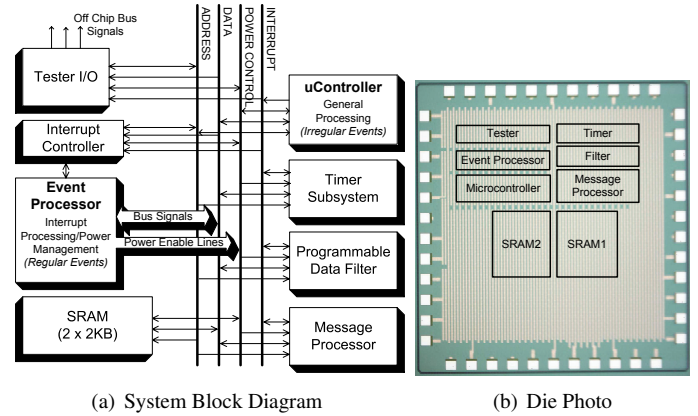
Mark Hempstead, Gu-Yeon Wei, David Brooks  
School of Engineering and Applied Sciences, Harvard University  
{mhempste, guyeon, dbrooks}@eecs.harvard.edu

We have implemented a system architecture for wireless sensor network nodes in 130nm CMOS. It operates at 550 mV and 12.5 MHz. Our system uses 100x less power when idle than a traditional microcontroller, and 10-600x less energy when active. It achieves energy efficiency by using a small event processor, heterogeneous hardware accelerators, and application-controlled VDD gating.

Networks of ultra-low-power nodes that include sensing, computation, and wireless communication have applications in medicine, science, industrial automation, and security. System-on-chip (SoC) implementations of such nodes can provide both energy efficiency and adequate performance to meet the long deployment lifetimes and bursts of computation that characterize wireless sensor network (WSN) applications. Proposed SoC for WSNs typically rely on general-purpose microcontrollers as the main compute engine and often run in subthreshold to minimize energy [4]. Unfortunately, subthreshold operation increases susceptibility to on-die parameter variations, limits the performance needed for real-time applications, and requires custom SRAM design [2]. In order to accommodate the wide variety of computing needs in WSNs while minimizing energy consumption, we propose an accelerator-based system architecture. Our design fully embraces the accelerator-based computing paradigm, including acceleration for the network layer (routing) and application layer (data filtering). Moreover, our architecture can disable the accelerators via VDD-gating to minimize leakage current during the long idle times common in WSN applications.

We target a class of habitat monitoring WSN applications that aim for long deployment lifetimes and incorporate data filtering and multihop routing on the nodes. Specifically, this architecture was informed by the volcano monitoring system deployed by Werner-Allen et al. [1]. In that system, nodes sampled both seismic and infrasound signals and use an exponentially weighted moving average (EWMA) filter to detect interesting events and transmit data back to a team of vulcanologists.

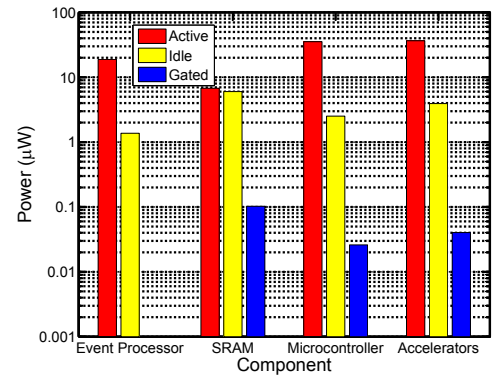
Fig. 1 (a) presents a block diagram of the prototype chip. The Event Processor (EP) is a small programmable state machine that runs interrupt service routines (ISRs) to control the flow of data between the on-chip memory and multiple accelerators, such as the message processor, programmable data filter, and timer, which are memory mapped and connected via the system bus [3]. The EP also acts as a power manager, turning accelerators on and off as needed by the running application. While the system also includes an 8-bit general-purpose microcontroller to handle infrequent and irregular tasks, it can usually be disabled. During long idle times, only the EP—and perhaps select blocks such as



**Figure 1. An Accelerator-based System for WSNs.**

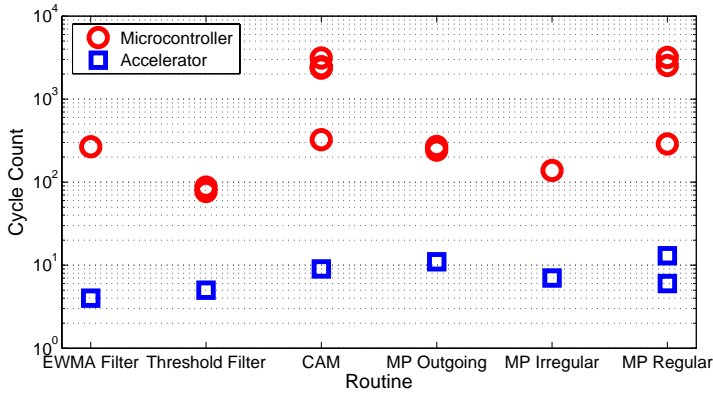
the timer—must be powered. The tester I/O block facilitates testing to verify functionality.

The chip was manufactured in a 130nm bulk CMOS process with eight layers of metal. A die photograph is shown in Figure 1 (b). The system contains 444,982 transistors including 4KB of foundry-supplied SRAM.

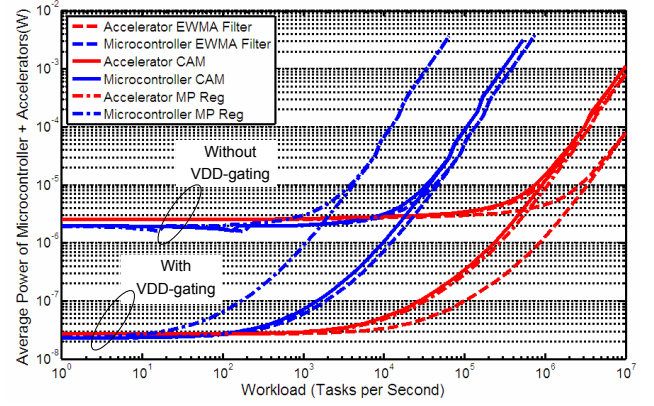


**Figure 2. Per Block Power Consumption 550 mV and 12.5 MHz.**

Our first experimental measurements have verified reliable operation across a range of lower clock frequencies—25 kHz to 12.5 MHz—that are suited to the low power needs of WSN applications. SRAM reliability limits the minimum operating voltage to 450mV. Fig. 2 plots the per-block power consumption of the system, running custom microbenchmarks written to exercise each block in three operating modes - active (12.5MHz @ 550mV), idle (0MHz @ 550mV), and powered off (VDD-gated). VDD-gating reduces the power consumption of individual blocks by 50-100x,



(a) Cycle Count Comparison



(b) Power vs. Workload

**Figure 3. Performance and Power Benefits of Specialization.**

which helps to minimize power consumption during long periods of inactivity. The event processor block cannot be VDD-gated since it must always be available to handle interrupts.

In this talk, we compare our prototype to nine processors for WSNs in the literature. Because the commonly used metric of energy-per-instruction cannot be easily applied to accelerator-based systems, we introduce the concept of *energy-per-task*. We defined a task as a collection of dependent computations that are executed periodically. We present measurements of a task that is similar to the volcano monitoring application. This task takes 131 cycles to execute and consumes 678.9 pJ at 550 mV and 12.5 MHz. An equivalent routine written for the Mica2 mote requires 1532 instructions. Using this information we compute the *energy per equivalent instruction* as 0.44 pJ, which is significantly lower than systems in the literature – the lowest energy systems, general purpose cores operating in subthreshold, consume 2-3 pJ per instruction.

This analysis does not isolate the benefits of an accelerator-based architecture from the process technology, circuit implementation, and amount of SRAM. Thus, we compare the cycle count and energy of full applications running on accelerators to running on the on-die general-purpose microcontroller. These applications combine data filtering, outgoing message preparation, and flood-based message routing, which are prototypical WSN routines. We analyze routines for data filtering (EWMA and threshold), network routing using a CAM structure, recording an outgoing message, detection of an incoming irregular message, and automatic relay of a regular message. The on-die Z80 microcontroller closely resembles 8-bit architectures employed in other WSN SoCs. For fairness, all routines were written in assembly and hand-tuned for accelerator- and microcontroller-based operation, respectively. Fig. 3 (a) presents the cycle count of each routine for both scenarios. Multiple points for a particular routine reflect different inputs that yield different performances. Accelerator implementations see cycle speedups from 15 to 635x, directly translating in energy savings. In this talk, we show, through measurements of energy consumption, that hardware accelerators consume 1/10th to 1/600th the energy consumed by software-based routines running on the microcontroller.

Building on individual characterizations above, we compare compute-block power consumption for different workload re-

quirements and include idle power in our analysis. These results exclude additional system power overheads (e.g., EP and SRAM) common to both types of systems in order to clarify the comparison. WSN workload intensity varies significantly depending on the observed phenomena—from 1 task/minute for weather observations to  $> 10^5$  tasks/second for high-frequency data collection.

Figure 3 (b) plots the average power consumption of routines run on either the accelerators or the microcontroller while varying workload intensity. For each datapoint, the lowest power voltage/frequency operating point was chosen. For light workloads ( $< 10$  tasks/sec), the system can operate at the lowest voltage and frequency (450mV, 25 KHz) and power consumption is dominated by leakage current. For medium-intensity workloads ( $10^4$  tasks/sec), using accelerators provides 1000x power savings due to a 635x speedup in cycle counts and a 50% lower supply voltage. As workload increases, active power dominates until the clock frequency required by the microcontroller reaches the performance limit of the system at the maximum supply voltage of 1.2V. Routines run on the accelerator can operate up to  $10^7$  tasks per second with a voltage less than 1.1 V. Also shown in the plot, VDD-gating lowers the power consumption for both scenarios under light loads, but the accelerators' higher inherent performance enables VDD-gating for longer periods of time that translate to additional power savings.

In conclusion, this accelerator-based system is well suited for both high performance and low performance sensor network applications. The system provides efficient computation through hardware acceleration for habitat monitoring applications. The modular architecture and event processor enable the management of idle power through VDD-gating.

## References

- [1] G. Werner-Allen et al. Fidelity and yield in a volcano monitoring sensor network. In *Symposium on Operating Systems Design and Implementation (OSDI)*, November 2006.
- [2] J. Kwong et al. A 65nm Sub- $V_t$  Microcontroller with Integrated SRAM and Switched-Capacitor DC-DC Converter. In *IEEE International Solid-State Circuits Conference (ISSCC)*, February 2008.
- [3] M. Hempstead et al. An ultra low power system architecture for sensor network applications. In *International Symposium on Computer Architecture (ISCA)*, June 2005.
- [4] S. Hanson et al. Exploring variability and performance in a sub-200-mv processor. *IEEE Journal of Solid-State Circuits*, 43(4):881–891, April 2008.